Galois embedding from universal types into existential types : Extended Abstract(Algorithmic problems in algebra, languages and computation systems)

AUTHOR(S):

Fujita, Ken-etsu

数理解析研究所講究録
1503 巻 2006 年 121-128

Galois embedding from universal types into existential types – Extended Abstract –

Ken-etsu Fujita (藤田 憲悦)
Department of Computer Science,
Gunma University (群馬大学)

Abstract

We show that there exist translations between polymorphic λ-calculus and a subsystem of minimal logic with existential types, which form a Galois insertion (embedding). From a programming point of view, this result means that abstract data types can interpret polymorphic functions under CPS-translation.

1 Introduction

We show that polymorphic types can be interpreted by the use of second order existential types. For this, we prove that there exist translations between polymorphic λ-calculus $\lambda 2$ and subsystem of minimal logic with existential types, which form a Galois connection and moreover a Galois insertion (embedding). From a programming point of view, this result means that abstract data types can interpret polymorphic functions under the so-called modified CPS-translation [Plot75, SF93].

Our main interest is a neat connection and proof duality between polymorphic types (2nd order universally quantified formulae) and existential types (2nd order existentially quantified formulae). It is logically quite natural like de Morgan's duality, and computationally still interesting, since dual of polymorphic functions with universal type can be regarded as abstract data types with existential type [MP85]. Instead of classical systems like [Pari92, Seli01, Wad03], even intuitionistic systems can enjoy that polymorphic types can be interpreted by existential types. That is, computationally polymorphic function with universal type $\forall X.A$ can be interpreted by abstract data types with existential type, such that the parametric polymorphic function $\lambda X.M$ for X can be viewed, under the so-called CPS-translation *, as an abstract data type for X, which is waiting for an implementation with type $\exists X.A^*$. This interpretation also contains proof duality, such that the universal formulae introduction rule is interpreted by the use of the existential formulae elimination rule, and the universal

elimination by the existential introduction. Moreover, we established not only a Galois connection but also a Galois insertion (embedding) from polymorphic λ-calculus (Girard-Reynolds) into a calculus with existential types. From the neat connection between the calculi, the fundamental properties such as normalization and Church-Rosser are related each other via CPS-translation and left adjoint.

2 Source calculus: $\lambda 2$

We first introduce our source calculus, 2nd order λ-calculus denoted by $\lambda 2$. For simplicity, we adopt its domain-free style.

Definition 1 (Types)

$$A ::= X \mid A \Rightarrow A \mid \forall X.A$$

Definition 2 ((Pseudo)$\lambda 2$-terms)

$$\Lambda 2 \ni M ::= x \mid \lambda x.M \mid \lambda x.M \mid MM \mid \lambda X.M \mid MA$$

Definition 3 (Reduction rules) (β) $(\lambda x.M_1)M_2 \rightarrow M_1[x := M_2]$

(η) $\lambda x.Mx \rightarrow M$, *if* $x \notin FV(M)$

(β_t) $(\lambda X.M)A \rightarrow M[X := A]$

(η_t) $\lambda X.MX \rightarrow M$, *if* $X \notin FV(M)$

$FV(M)$ denotes a set of free variables in M.

3 Target calculus: $\lambda^{\exists}$

We next define our target calculus denoted by $\lambda^{\exists}$, which is logically a subsystem of minimal logic consisting of constant $\bot$, negation, conjunction and 2nd order existential quantification.

Definition 4 (Types)

$$A ::= \bot \mid X \mid \neg A \mid A \wedge A \mid \exists X.A$$

Definition 5 ((Pseudo)$\lambda^{\exists}$-terms)

$$\begin{aligned} \Lambda^{\exists} \ni M \quad ::= \quad & x \mid \lambda x.M \mid MM \mid \langle M, M \rangle \mid \texttt{let}\ \langle x, x \rangle = M\ \texttt{in}\ M \\ & \mid \langle A, M \rangle \mid \texttt{let}\ \langle X, x \rangle = M\ \texttt{in}\ M \end{aligned}$$

Definition 6 (Reduction rules) (β) $(\lambda x.M_1)M_2 \rightarrow M_1[x := M_2]$

(η) $\lambda x.Mx \rightarrow M$, *if* $x \notin FV(M)$

$(\mathtt{let}_\wedge)$ $\mathtt{let}\ \langle x_1, x_2\rangle = \langle M_1, M_2\rangle\ \mathtt{in}\ M \to M[x_1 := M_1, x_2 := M_2]$

$(\mathtt{let}_{\wedge_\eta})$ $\mathtt{let}\ \langle x_1, x_2\rangle = M_1\ \mathtt{in}\ M[z := \langle x_1, x_2\rangle] \to M[z := M_1]$, *if* $x_1, x_2 \notin FV(M)$

$(\mathtt{let}_\exists)$ $\mathtt{let}\ \langle X, x\rangle = \langle A, M_1\rangle\ \mathtt{in}\ M \to M[X := A, x := M_1]$

$(\mathtt{let}_{\exists_\eta})$ $\mathtt{let}\ \langle X, x\rangle = M_1\ \mathtt{in}\ M[z := \langle X, x\rangle] \to M_2[z := M_1]$, *if* $X, x \notin FV(M_2)$

We write simply $(\mathtt{let})$ for either $(\mathtt{let}_\wedge)$ or $(\mathtt{let}_\exists)$, and $(\mathtt{let}_\eta)$ for $(\mathtt{let}_{\wedge_\eta})$ or $(\mathtt{let}_{\exists_\eta})$.

4 CPS-translation * from $\Lambda 2$ into $\Lambda^\exists$

We define a translation so-called modified CPS-translation * from pseudo $\lambda 2$-terms into pseudo $\lambda^\exists$-terms, which preserves not only reduction relation but also typing relation introduced later. In each case, a fresh and free variable a is introduced, which is called a continuation variable.

Definition 7 *1.* $x^* = xa$

2. $(\lambda x.M)^* = \mathtt{let}\ \langle x, a\rangle = a\ \mathtt{in}\ M^*$

3. $(M_1 M_2)^* = \begin{cases} M_1^*[a := \langle x, a\rangle] & \text{for } M_2 \equiv x \\ M_1^*[a := \langle \lambda a.M_2^*, a\rangle] & \text{otherwise} \end{cases}$

4. $(\lambda X.M)^* = \mathtt{let}\ \langle X, a\rangle = a\ \mathtt{in}\ M^*$

5. $(MA)^* = M^*[a := \langle A^*, a\rangle]$

6. $X^* = X$; $(A_1 \Rightarrow A_2)^* = \neg A_1^* \wedge A_2^*$; $(\forall X.A)^* = \exists X.A^*$

Remarked that M^* contains exactly one free occurrence of a continuation variable a, and M^* has neither β-redex nor η-redex. Let $\lambda X.M$ have type $\forall X.A$. Then, under the translation, parametric polymorphic function $(\lambda X.M)^*$ with respect to X becomes an abstract data type for X, which is waiting for an implementation with type $\exists X.A^*$ together with an interface (a signature) with type A^*, i.e.,

$$\mathtt{abstype}\ X\ \mathtt{with}\ a{:}A^*\ \mathtt{is}\ a\ \mathtt{in}\ M^*$$

in a familiar notation.

Lemma 1 *1. We have* $M_1^*[x := \lambda a.M_2^*] \to_{\beta\eta}^* (M_1[x := M_2])^*$.

In particular, $M_1^*[x := \lambda a.M_2^*] \to_\beta^* (M_1[x := M_2])^*$ *provided that* M_2 *is not a variable.*

2. If $M_1 \to_\beta M_2$, *then* $M_1^* \to_{\beta\eta\mathtt{let}}^+ M_2^*$.

3. If $M_1 \to_\eta M_2$, *then* $M_1^* \to_{\eta\mathtt{let}_\eta}^+ M_2^*$

For inverse translation, the following mutual induction defines *Univ* and $\mathcal{C}$, respectively for denotations and for continuations.

$$\frac{C_a \in \mathcal{C}}{xC_a \in \mathit{Univ}} \qquad \frac{C_a \in \mathcal{C} \quad P \in \mathit{Univ}}{(\lambda a.P)C_a \in \mathit{Univ}}$$

$$\frac{C_a \in \mathcal{C} \quad P \in \mathit{Univ}}{\texttt{let } \langle x, a\rangle = C_a \texttt{ in } P \in \mathit{Univ}} \qquad \frac{C_a \in \mathcal{C} \quad P \in \mathit{Univ}}{\texttt{let } \langle X, a\rangle = C_a \texttt{ in } P \in \mathit{Univ}}$$

$$a \in \mathcal{C} \qquad \frac{C_a \in \mathcal{C}}{\langle x, C_a\rangle \in \mathcal{C}}$$

$$\frac{C_a \in \mathcal{C} \quad P \in \mathit{Univ}}{\langle \lambda a.P, C_a\rangle \in \mathcal{C}} \qquad \frac{C_a \in \mathcal{C}}{\langle A^*, C_a\rangle \in \mathcal{C}}$$

We write $\langle R_1, R_2, \ldots, R_n\rangle$ for $\langle R_1, \langle R_2, \ldots, R_n\rangle\rangle$ with $n > 1$, and $\langle R_1\rangle$ for R_1 with $n = 1$. $C_a \in \mathcal{C}$ is in the form of $\langle R_1, \ldots, R_n, a\rangle$ where R_i $(1 \leq i \leq n)$ is x, $\lambda a.P$, or A^* with $n \geq 0$.

According to the general property of Galois connection, for $P \in \mathit{Univ}$, an upper adjoint (left adjoint) $\sharp$ can be defined as follows, where a preorder $\sqsubseteq$ is defined by $\rightarrow^*$, the reflexive and transitive closure of one step reduction $\rightarrow$.

$P^\sharp \stackrel{\text{def}}{=} \sup\{M \in \Lambda 2 \mid M^* \sqsubseteq P\}$ where $P \sqsubseteq Q$ iff $Q \rightarrow^* P$.

In fact, this definition works well, which can be verified by case analysis on $P \in \mathit{Univ}$, in the following recursive way:

- Case $P \equiv xC \equiv x\langle R_1, \ldots, R_n, a\rangle$ with $n \geq 0$

  From the definition of *, $P^\sharp$ is in the form of $xM_1 \ldots M_n$ for some M_i, where

  - If $R_i \equiv x_i$, then $M_i \equiv x_i$.
  - If $R_i \equiv \lambda a.P_i$, then find similarly M_i such that $M_i^* \sqsubseteq P_i$.
  - If $R_i \equiv A_i^*$, then $M_i \equiv A_i$.

- Case $P \equiv (\lambda a.P')C$

  We have no M such that $M^* \equiv (\lambda a.P')C$. Then we should fine M such that $M^* \sqsubseteq P'[a := C] \sqsubseteq (\lambda a.P')C$, where a is a linear variable.

- Case $P \equiv \texttt{let } \langle x, a\rangle = C \texttt{ in } P'$ with $C = \langle R_1, \ldots, R_n, a\rangle$ $(n \geq 0)$

  $P^\sharp$ is in the form of $(\lambda x.M)N_1 \ldots N_n$ for some M and N_i, where we should find M such that $M^* \sqsubseteq P'$, and:

  - If $R_i \equiv x_i$ then $N_i \equiv x_i$.
  - If $R_i \equiv \lambda a.P_i$ then find N_i such that $N_i^* \sqsubseteq P_i$.
  - If $R_i \equiv A_i^*$ then $N_i \equiv A_i$.

Here we have a valid induction measure, since continuation variable is linear and we always choose strictly smaller subterms to find an upper adjoint. This definition $\sharp$ is summarized inductively as follows, where we write $C[\ ]$ for $C \in \mathcal{C}$ with a hole $[\ ]$:

Definition 8 *0.* $x^\sharp = x$; $(\lambda a.P)^\sharp = P^\sharp$; $(A^*)^\sharp = A$

1. $(xC)^\sharp = C^\sharp[x^\sharp]$
2. $((\lambda a.P)C)^\sharp = C^\sharp[(\lambda a.P)^\sharp]$
3. $(\mathtt{let}\ \langle x, a\rangle = C\ \mathtt{in}\ P)^\sharp = C^\sharp[\lambda x.P^\sharp]$
4. $(\mathtt{let}\ \langle X, a\rangle = C\ \mathtt{in}\ P)^\sharp = C^\sharp[\lambda X.P^\sharp]$
5. $a^\sharp = [\ ]$
6. $\langle x, C\rangle^\sharp = C^\sharp[[\ ]x^\sharp]$
7. $\langle \lambda a.P, C\rangle^\sharp = C^\sharp[[\ ](\lambda a.P)^\sharp]$
8. $\langle A^*, C\rangle^\sharp = C^\sharp[[\ ](A^*)^\sharp]$

Note that $C_a^\sharp = [\ ]R_1^\sharp \ldots R_n^\sharp$ with left associativity, if $C_a \in \mathcal{C}$ is in the form of $\langle R_1, \ldots, R_n, a\rangle$. We also remark that if we have $P[a := C]$, then there uniquely exists a free occurrence of a in $P \in \mathit{Univ}$.

Lemma 2 *1.* $(P[a := C])^\sharp \equiv C^\sharp[P^\sharp]$

2. *Let* $P, P_0 \in \mathit{Univ}$ *and* $C \in \mathcal{C}$.
   $(P[x := \lambda a.P_0])^\sharp = P^\sharp[x := P_0^\sharp]$
   $(C[x := \lambda a.P_0])^\sharp = C^\sharp[x := P_0^\sharp]$

Proposition 1 *Let* $P_1, P_2 \in \mathit{Univ}$.

1. *If* $P_1 \to_\beta P_2$, *then* $P_1^\sharp \equiv P_2^\sharp$.
2. *If* $P_1 \to_\eta P_2$, *then* $P_1^\sharp \equiv P_2^\sharp$.
3. *If* $P_1 \to_{\mathtt{let}} P_2$, *then* $P_1^\sharp \to_\beta P_2^\sharp$.
4. *If* $P_1 \to_{\mathtt{let}_\eta} P_2$, *then* $P_1^\sharp \to_\eta P_2^\sharp$.

Proposition 2 *Let* $M \in \Lambda 2$ *and* $P \in \mathit{Univ}$.

1. $M^{*\sharp} \equiv M$ *and* $P \to_{\beta\eta}^* P^{\sharp *}$
2. *If* M *is in* $\lambda 2$*-normal, then* M^* *is in* $\lambda^\exists$*-normal.*
   If P *is in* $\lambda^\exists$*-normal, then* $P^\sharp$ *is in* $\lambda 2$*-normal.*

Theorem 1 (Galois insertion) *$\langle \Lambda 2, Univ, *, \sharp \rangle$ forms a Galois connection, in particular Galois insertion such that $M^{*\sharp} \equiv M$. That is, let $M, M_1, M_2 \in \Lambda 2$ and $P, P_1, P_2 \in Univ$. Then we have the following properties:*

1. *If $M_1 \to^*_{\lambda 2} M_2$ then $M_1^* \to^*_{Univ} M_2^*$.*
2. *If $P_1 \to^*_{Univ} P_2$ then $P_1^\sharp \to^*_{\lambda 2} P_2^\sharp$.*
3. *$M^{*\sharp} \to^*_{\lambda 2} M$ and $P \to^*_{Univ} P^{\sharp *}$.*

In other words:

$$P \to^*_{Univ} M^* \text{ if and only if } P^\sharp \to^*_{\lambda 2} M$$

Corollary 1 1. *Strong normalization of Univ implies that of $\lambda 2$.*

2. *$\lambda 2$ is weakly normalizing iff Univ is weakly normalizing.*
3. *There exists a one-to-one correspondence between $\lambda 2$-normal forms and Univ-normal forms.*
4. *$\lambda 2$ is Church-Rosser iff Univ is Church-Rosser.*

   We remark that $\lambda^\exists$ itself is NOT Church-Rosser.
5. *Let $\downarrow P \stackrel{\text{def}}{=} \{Q \in Univ \mid P \to^*_{\lambda^\exists} Q\}$ for $P \in Univ$. Then an inverse image of $\downarrow P$ is principal, in the sense that the inverse image of $\downarrow P$ is equal to $\downarrow(P^\sharp)$, that is, generated by $P^\sharp$.*
6. *Let $\downarrow_{\lambda^\exists} [\![\Lambda 2]\!] \stackrel{\text{def}}{=} \{P \mid M^* \to^*_{\lambda^\exists} P \text{ for some } M \in \Lambda 2\}$.*

   Let $\uparrow_{\beta\eta} [\![\Lambda 2]\!] \stackrel{\text{def}}{=} \{P \in Univ \mid P \to^*_{\beta\eta} M^* \text{ for some } M \in \Lambda 2\}$.

   Then we have $\downarrow_{\lambda^\exists} [\![\Lambda 2]\!] \subseteq \uparrow_{\beta\eta} [\![\Lambda 2]\!] = Univ$.

5 Proof duality

Finally, we give sets of type assignment rules for $\lambda 2$ and $\lambda^\exists$, respectively, as follows.

$\lambda 2$:

$$\frac{x{:}A \in \Gamma}{\Gamma \vdash x : A}$$

$$\frac{\Gamma, x{:}A_1 \vdash M : A_2}{\Gamma \vdash \lambda x{:}A_1.M : A_1 \Rightarrow A_2}\ (\Rightarrow I) \qquad \frac{\Gamma \vdash M_1 : A_1 \Rightarrow A_2 \quad \Gamma \vdash M_2 : A_1}{\Gamma \vdash M_1 M_2 : A_2}\ (\Rightarrow E)$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \lambda X.M : \forall X.A}\ (\forall I)^* \qquad \frac{\Gamma \vdash M : \forall X.A}{\Gamma \vdash M A_1 : A[X := A_1]}\ (\forall E)$$

where $(\forall I)^*$ denotes the eigenvariable condition $X \notin FV(\Gamma)$.

$\lambda^{\exists}$:

$$\frac{x{:}A \in \Gamma}{\Gamma \vdash x : A}$$

$$\frac{\Gamma, x{:}A \vdash M : \bot}{\Gamma \vdash \lambda x{:}A.M : \neg A}\ (\neg I) \qquad \frac{\Gamma \vdash M_1 : \neg A \quad \Gamma \vdash M_2 : A}{\Gamma \vdash M_1 M_2 : \bot}\ (\neg E)$$

$$\frac{\Gamma \vdash M_1 : A_1 \quad \Gamma \vdash M_2 : A_2}{\Gamma \vdash \langle M_1, M_2 \rangle : A_1 \wedge A_2}\ (\wedge I) \qquad \frac{\Gamma \vdash M_1 : A_1 \wedge A_2 \quad \Gamma, x_1{:}A_1, x_2{:}A_2 \vdash M : A}{\Gamma \vdash \mathtt{let}\ \langle x_1, x_2 \rangle = M_1\ \mathtt{in}\ M : A}\ (\wedge E)$$

$$\frac{\Gamma \vdash M : A[X := A_1]}{\Gamma \vdash \langle A_1, M \rangle_{\exists X.A} : \exists X.A}\ (\exists I) \qquad \frac{\Gamma \vdash M : \exists X.A \quad \Gamma, x{:}A \vdash M_1 : A_1}{\Gamma \vdash \mathtt{let}\ \langle X, x \rangle = M\ \mathtt{in}\ M_1 : A_1}\ (\exists E)^*$$

where $(\exists E)^*$ denotes the eigenvariable condition $X \notin FV(\Gamma, A_1)$.

Proposition 3 $\Gamma \vdash_{\lambda 2} M : A$ *if and only if* $\neg\Gamma^*, a{:}A^* \vdash_{\lambda^{\exists}} M^* : \bot$

Theorem 2 (Proof duality) *Let* Π *be a normal deduction of* $\Gamma \vdash_{\lambda 2} M : A$, *and in* Π, *let* π *be a path:*

$A_1(E_1)A_2(E_2)\ldots A_i(E_i)A_{i+1}(I_{i+1})\ldots A_{n-1}(I_{n-1})A_n$.

Then, in the deduction of $\neg\Gamma^*, a : A^* \vdash_{\lambda^{\exists}} M^* : \bot$, *there exists a path* π^* *as follows:*

$A_n^*(I_{n-1})^*A_{n-1}^*\ldots(I_{i+1})^*A_{i+1}^*(E_i)^*A_i^*\ldots(E_2)^*A_2^*(E_1)^*A_1^*$

under the following correspondence:

$(\Rightarrow I)^* = (\wedge E)$; $(\Rightarrow E)^* = (\wedge I)$; $(\forall I)^* = (\exists E)$; $(\forall E)^* = (\exists I)$.

References

[Fuji03] K. Fujita: A sound and complete CSP-translation for $\lambda\mu$-Calculus, Lecture Notes in Computer Science 2701, pp. 120–134, 2003.

[Fuji05] K. Fujita: Galois embedding from polymorphic types into existential types, Lecture Notes in Computer Science 3461, pp. 194–208, 2005.

[MP85] J. C. Mitchell and G. D. Plotkin: Abstract types have existential type, *Proc. the 12th Annual ACM Symposium on Principles of Programming Languages*, pp. 37–51, 1985.

[Pari92] M. Parigot: $\lambda\mu$-Calculus: An Algorithmic Interpretation of Classical Natural Deduction, Lecture Notes in Computer Science 624, pp. 190–201, 1992.

[Plot75] G. Plotkin: Call-by-Name, Call-by-Value and the λ-Calculus, *Theoretical Computer Science*, Vol. 1, pp. 125–159, 1975.

[Pra65] D. Prawitz: *NATURAL DEDUCTION, A Proof Theoretical Study*, ALMQVIST&WIKSELL, Stockholm, 1965.

[Seli01] P. Selinger: Control Categories and Duality: on the Categorical Semantics of the Lambda-Mu Calculus, *Math. Struct. in Compu. Science*, Vol. 11, pp. 207–260, 2001.

[SF93] A. Sabry and M. Felleisen: Reasoning about Programs in Continuation-Passing Style, *LISP AND SYMBOLIC COMPUTATION: An International Journal*, Vol. 6, pp. 289–360, 1993.

[Wad03] Ph. Wadler: Call-by-value is dual to call-by-name, *International Conference on Functional Programming*, August 25-29, Uppsala, 2003.